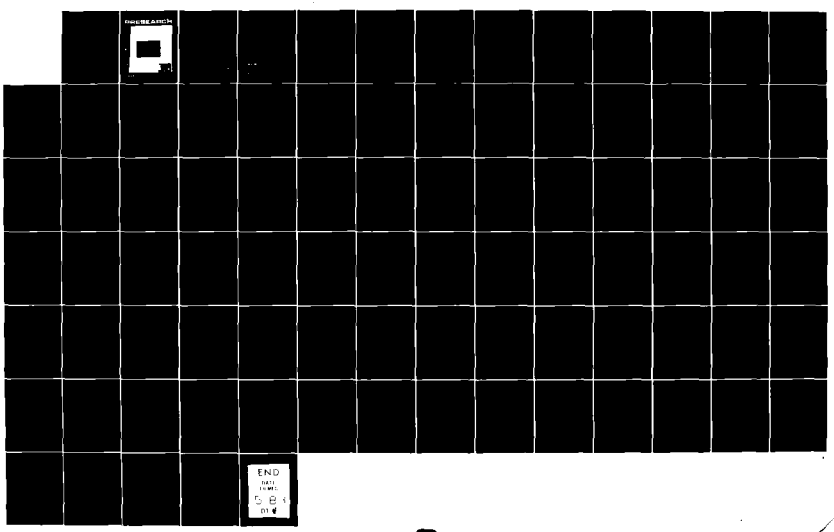
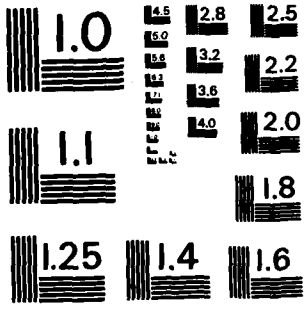
AD-A127 208 FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION VOLUME 7 - 1/1
SOURCE LISTINGS: TABLE AND GRAPHS(U) PRESEARCH INC
ARLINGTON VA DEC 82 FPO-1-82-(38) N62477-81-C-0025
UNCLASSIFIED F/G 9/2 NL





PRESEARCH

FLEET MOORING LEG
DESIGN PROGRAM DOCUMENTATION

Volume 7

SOURCE LISTINGS:
TABLE AND GRAPHS

FPO-1-82-(38)

December 1982

General Distribution

PRESEARCH INCORPORATED

2361 S. JEFFERSON DAVIS HIGHWAY, ARLINGTON, VA. 22202 (703) 553-2700

PRESEARCH INCORPORATED

(1)

FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION

Volume 7

SOURCE LISTINGS: TABLE AND GRAPHS

FPO-1-82-(38)

December 1982

General Distribution

Performed for
Ocean Engineering and Construction Project Office
Chesapeake Division
Naval Facilities Engineering Command
Washington, D.C. 20374
Under
Contract N62477-81-C-0025

Presearch Incorporated
2361 South Jefferson Davis Highway
Arlington, Virginia 22202

FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION

Volume 7

SOURCE LISTINGS: TABLE AND GRAPHS

IV. SOURCE LISTINGS

TABLE AND GRAPHS

GRAPH1 – MOOR$5, PAGES 222-303

```
e! sys final/t2for/graph1 for##
      subroutine graph1
c
c Produce annotated symbolic depiction of compound leg, tables of
c parameters input to define it, and tables of computed forces,
c coordinates, angles and tensions
c
      implicit integer*2 (#)
c
c COMMON BLOCK DECLARATIONS
c
c
c     TITLES
c
      integer*1       ctitle(114)
      common /titles/ ctitle
c
c     DATIME
c
      integer*1       cdatim(16)
      common /datime/ cdatim
c
c     VARIN
c
      integer*1       cvarin(172)
      common /varin/ cvarin
c
c     VAROUT
c
      integer*1       cvarol(240),cvaro2(100)
      common /varout/ cvarol,     cvaro2
c
c     VARG
c
      integer*1     cvarg(240)
      common /varg/ cvarg
c
c     UNKNOW
c
      integer*1       cunkno(12)
      common /unknow/ cunkno
c
c     GROPT
```

```
c
      integer*1       cgrop1(44)
      common /grop1/ cgrop1
c
c     GRP2CN
c
      integer*1       cgrp21(218),cgrp22(82)
      common /grp2cn/ cgrp21,     cgrp22
c
c PRINT TABLES OF INPUT PARAMETERS AND COMPUTED VALUES
c
      call RWCOM1(1)
      call ovlink('GRIN1 ')
      call ovlink('GRIN2 ')
      call ovlink('GROUT1 ')
      call ovlink('GROUT2 ')
      call RWCOM1(2)
      return
      end
*
```

```
e! sys final/i2for/grini for##
      subroutine grini
c
c Print Graph 1 header, legend of units, and input parameter list
c
      implicit integer*2 (*)
c
c COMMON BLOCK DECLARATIONS
c
c     LUNITS
c
      integer*2       screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /lunits/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &    pref1,dum1,ext1,ext2
c
c     TITLES
c
      integer*1 title(50),ifile(32),ofile(32)
      common /titles/ title,ifile,ofile
c
c     DATIME
c
      integer*2       idate(5),ihour,imin,isec
      common /datime/ idate,   ihour,imin,isec
c
c     VARIN
c
      integer*2 illeg,list
      integer*4 nnca,nncb
      real angla,anglb,
     &    scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
     &    scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &    scop3a,scop3b,wgt3a,wgt3b,
     &    slip,frict,                clmp3,
     &    scop4,        wgt4,
     &    anksep,
     &    p1x,p1z,p1d,
     &    p2x,p2z,p2d,
     &    p3x,p3z,p3d,
     &    hload,hdir,
     &    rbuoy,xbuoy,zbuoy,
     &    depth0,pdir
```

```
      common /varin/ iileg,iist,
     &     nnca,nncb,
     &     angla,anglb,
     &     scopla,scoplb,wgtla,wgtlb,clmpla,clmplb,
     &     scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &     scop3a,scop3b,wgt3a,wgt3b,
     &     slip,fricl,                clmp3,
     &     scop4,         wgt4,
     &     anksep,
     &     plx,plz,pld,
     &     p2x,p2z,p2d,
     &     p3x,p3z,p3d,
     &     hload,hdir,
     &     rbuoy,xbuoy,zbuoy,
     &     depth0,pdir
c
c     GCB
c
      integer*2 gbuff(24),lugraf,luptfl,ludbug
      common /gcb/ gbuff ,lugraf,luptfl,ludbug
c
c LOCAL VARIABLES
c
      integer*1 legnm(23,3)
      integer*2 rdate
      integer*2 three,five
      integer*2 funkey
c
c DATA INITIALIZATION
c
      data three,five/3,5/
      data legnm/'simple                  ','compound - equalizer    ',
     & 'compound - spider plate'/

c
c EXECUTABLE PORTION
c
c HEADER
c
      call gfinit
      call date(rdate)
      call undate(rdate,idate)
      call time(ihour,imin,isec)
```

```
      write(screen,1005) ihour,imin,isec
 1005 format(1x,'SOLUTION COMPLETED AT ',i2,' ',i2,' ',i2)
      call readfk(funkey)
      call erase
      call chrsiz(three)
      write(screen,1010)idate,ihour,imin,isec
 1010 format(1x,'Date    ',5a2,38x,'SUMMARY',37x,
     & 'Time    ',i2,' ',i2,' ',i2,)
      write(screen,1011)
 1011 format('+','                                                ')
      call chrsiz(five)
      write(screen,1020)title
 1020 format(1x,18x,50a1)
      write(screen,1030)
 1030 format(1x,'INPUT ')
      write(screen,1011)
c FILE NAMES
      call chrsiz(three)
      write(screen,1040)ifile,ofile
 1040 format('+',12x,'Original Input From File ',32a1,
     & ' Revised to File ',32a1)
c UNITS
      write(screen,1050)
 1050 format(1x,13x,'Angles          - Degrees'/
     & 1x,13x,'Distances       - Feet'/
     & 1x,5x,'Units   Linear Weights - Pounds/Foot'/
     & 1x,13x,'Weights         - Kilopounds'/
     & 1x,13x,'Forces          - Kilopounds')
c LEG TYPE
      write(screen,1110)legnm(1,ileg) 23
 1110 format(1x,'LEG '/
     & 1x,2x,'Type    ',23a1,'---A--- ---B---')
c ANCHOR SEPARATION
      write(screen,1120)
 1120 format(1x,2x,'Anchor Separation')
      if (anksep ne 9999.99)write(screen,1121)anksep
 1121 format('+',36x,f7.2)
c SEGMENTS IN BRANCH
      write(screen,1130)nnca
 1130 format(1x,2x,'Segments in Branch',15x,i1)
      if (ileg ne 1)write(screen,1131)nncb
 1131 format('+',43x,i1)
c ANGLE TO BOTTOM
```

```
      write(screen,1140)
 1140 format(1x,2x,'Angle to Bottom')
      if (angle ne 9999 99)write(screen,1141)angle
      if (anglb ne 9999 99)write(screen,1142)anglb
 1141 format('+',32x,f7 2)
 1142 format('+',40x,f7 2)
c LENGTH OF SEGMENT 1
      write(screen,1150)
 1150 format(1x,2x,'Length of Segment 1',8x,'S1')
      write (screen,1141)scopla
      if (scoplb ne 9999 99)write(screen,1142)scoplb
c LINEAR WEIGHT OF SEGMENT 1
      write(screen,1160)
 1160 format(1x,2x,'Linear Weight of Segment 1',1x,'W1')
      write(screen,1141)wgtla
      if (wgtlb ne 9999 99)write(screen,1142)wgtlb
c WEIGHT OF SINKER 1
      write(screen,1170)
 1170 format(1x,2x,'Weight of Sinker 1',9x,'C1')
      if (clmpla ne 9999 99)write(screen,1141)clmpla
      if (clmplb ne 9999 99)write(screen,1142)clmplb
c LENGTH OF SEGMENT 2
      write(screen,1180)
 1180 format(1x,2x,'Length of Segment 2',8x,'S2')
      if (scop2a ne 9999 99)write(screen,1141)scop2a
      if (scop2b ne 9999 99)write(screen,1142)scop2b
c LINEAR WEIGHT OF SEGMENT 2
      write(screen,1190)
 1190 format(1x,2x,'Linear Weight of Segment 2',1x,'W2')
      if (wgt2a ne 9999 99)write(screen,1141)wgt2a
      if (wgt2b ne 9999 99)write(screen,1142)wgt2b
c END
      return
      end
*
```

```
et sys final/12for/grin2 for##
      subroutine grin2
c
c Print Graph I input parameter list
c
      implicit integer*2 (*)
c
c COMMON BLOCK DECLARATIONS
c
c     LUNITS
c
      integer*2       screen,keybd,lul,lu2,niv99,siz99,ncpl
      integer*1 prefl(2),duml,exil(4),exi2(4)
      common /lunits/ screen,keybd,lul,lu2,niv99,siz99,ncpl,
     &    prefl,duml,exil,exi2
c
c     VARIN
c
      integer*2 illeg,list
      integer*4 nnca,nncb
      real angla,anglb,
     &    scopla,scoplb,wglla,wgllb,clmpla,clmplb,
     &    scop2a,scop2b,wgl2a,wgl2b,clmp2a,clmp2b,
     &    scop3a,scop3b,wgl3a,wgl3b,
     &    slip,frict,                clmp3,
     &    scop4,        wgl4,
     &    anksep,
     &    plx,plz,pld,
     &    p2x,p2z,p2d,
     &    p3x,p3z,p3d,
     &    hload,hdir,
     &    rbuoy,xbuoy,zbuoy,
     &    depths,pdir
      common /varin/ illeg,list,
     &    nnca,nncb,
     &    angla,anglb,
     &    scopla,scoplb,wglla,wgllb,clmpla,clmplb,
     &    scop2a,scop2b,wgl2a,wgl2b,clmp2a,clmp2b,
     &    scop3a,scop3b,wgl3a,wgl3b,
     &    slip,frict,                clmp3,
     &    scop4,        wgl4,
     &    anksep,
     &    plx,plz,pld,
```

```
     &     p2x ,p2z ,p2d ,
     &     p3x ,p3z ,p3d ,
     &     hload ,hdir ,
     &     rbuoy ,xbuoy ,zbuoy ,
     &     depth0 ,pdir
c
c      VAROUT
c
      real ola ,olb ,ol ,
     &     oha ,ohb ,oh ,
     &     ox1a ,ox3a ,ox5a ,ox1b ,ox3b ,ox5b ,ox7 ,ox8 ,
     &     oy1a ,oy3a ,oy5a ,oy1b ,oy3b ,oy5b ,oy7 ,oy8 ,
     &     oz1a ,oz3a ,oz5a ,oz1b ,oz3b ,oz5b ,oz7 ,oz8 ,
     &     oa1a ,oa2a ,oa3a ,oa4a ,oa5a ,oa6a ,
     &     oa1b ,oa2b ,oa3b ,oa4b ,oa5b ,oa6b ,oa7 ,oa8 ,
     &     ov1a ,ov2a ,ov3a ,ov4a ,ov5a ,ov6a ,
     &     ov1b ,ov2b ,ov3b ,ov4b ,ov5b ,ov6b ,ov7 ,ov8 ,
     &     ot1a ,ot2a ,ot3a ,ot4a ,ot5a ,ot6a ,
     &     ot1b ,ot2b ,ot3b ,ot4b ,ot5b ,ot6b ,ot7 ,ot8 ,
     &     odo ,oda ,odb ,
     &     oaf ,oafdir ,oafa ,oadir ,oafb ,obdir ,
     &     oslp ,oco1a ,oco1b
      integer*2 oisol ,obrnch
      common /varout/ ola ,olb ,ol ,
     &     oha ,ohb ,oh ,
     &     ox1a ,ox3a ,ox5a ,ox1b ,ox3b ,ox5b ,ox7 ,ox8 ,
     &     oy1a ,oy3a ,oy5a ,oy1b ,oy3b ,oy5b ,oy7 ,oy8 ,
     &     oz1a ,oz3a ,oz5a ,oz1b ,oz3b ,oz5b ,oz7 ,oz8 ,
     &     oa1a ,oa2a ,oa3a ,oa4a ,oa5a ,oa6a ,
     &     oa1b ,oa2b ,oa3b ,oa4b ,oa5b ,oa6b ,oa7 ,oa8 ,
     &     ov1a ,ov2a ,ov3a ,ov4a ,ov5a ,ov6a ,
     &     ov1b ,ov2b ,ov3b ,ov4b ,ov5b ,ov6b ,ov7 ,ov8 ,
     &     ot1a ,ot2a ,ot3a ,ot4a ,ot5a ,ot6a ,
     &     ot1b ,ot2b ,ot3b ,ot4b ,ot5b ,ot6b ,ot7 ,ot8 ,
     &     odo ,oda ,odb ,
     &     oaf ,oafdir ,oafa ,oadir ,oafb ,obdir ,
     &     oslp ,oco1a ,oco1b ,
     &     oisol ,obrnch
c
c      VARC
c
      double precision lla ,llb ,ll ,
     &     tana ,tanb ,tanr ,
```

```
     &     xx1a,xx3a,xx5a,xx3b,xx5b,xx7,xx8,
     &     ga11,ga12,ga21,ga22,ga31,ga32,
     &     gb11,gb12,gb21,gb22,gb31,gb32,
     &     g1,g2,
     &     xfa,xfb,xf
      common /varg/ lla,llb,ll,
     &     tana,tanb,tanr,
     &     xx1a,xx3a,xx5a,xx3b,xx5b,xx7,xx8,
     &     ga11,ga12,ga21,ga22,ga31,ga32,
     &     gb11,gb12,gb21,gb22,gb31,gb32,
     &     g1,g2,
     &     xfa,xfb,xf
c
c LOCAL VARIABLES
c
      integer*2 i
      integer*2 gbuff(24)
      real obmag,xxproj
c
c EXECUTABLE PORTION
c
c LINE 22
c     WEIGHT OF SINKER 2
      write(screen,110)
  110 format(1x,2x,'Weight of Sinker 2',9x,'C2')
      if (clmp2a ne 9999 99)write(screen,111)clmp2a
      if (clmp2b ne 9999 99)write(screen,112)clmp2b
  111 format('+',32x,f7 2)
  112 format('+',40x,f7 2)
c     OCEAN BOTTOM
      write(screen,120)
  120 format('+',48x,'OCEAN BOTTOM ')
c     OCEAN SURFACE
      write(screen,130)
  130 format('+',83x,'OCEAN SURFACE')
c LINE 23
C     START LENGTH OF SEGMENT 3
      write(screen,210)
  210 format(1x,2x,'Start Length of Segment 3',2x,'S3')
      if (scop3a ne 9999 99)write(screen,111)scop3a
      if (scop3b ne 9999 99)write(screen,112)scop3b
c     FLOOR DIRECTION
      write(screen,220)oafdir
```

```
  220 format('+',50x,'Floor Direction',2x,f7 2)
c LINE 24
c     LINEAR WEIGHT OF SEGMENT 3
      write(screen,310)
  310 format(1x,2x,'Linear Weight of Segment 3',1x,'W3')
      if (iileg eq 3  and  wgt3a ne 9999 99)write(screen,111)wgt3a
      if (iileg eq 3  and  wgt3b ne 9999 99)write(screen,112)wgt3b
      if (iileg ne 3  and  wgt3a ne 9999 99)write(screen,311)wgt3a
  311 format('+',36x,f7 2)
c     FLOOR SLOPE
      write(screen,320)aaf
  320 format('+',50x,'Floor Slope',6x,f7 2)
c     LOAD DIRECTION
      write(screen,330)
  330 format('+',85x,'Load Direction')
      if (hdir ne 9999 99)write(screen,331)hdir
  331 format('+',106x,f7 2)
c LINE 25
c     FRICTION COEFFICIENT
      write(screen,410)
  410 format(1x,2x,'Friction Coefficient')
      if (fric1 ne 9999 99)write(screen,311)fric1
c     X-DEPTH-Z HEADER
      write(screen,420)
  420 format('+',59x,'---X--- -Depth- ---Z---')
c     HORIZONTAL LOAD
      write(screen,430)
  430 format('+',85x,'Horizontal Load H')
      if (hload ne 9999 99)write(screen,331)hload
c LINE 26
c     WEIGHT OF EQUALIZER/SPIDER PLATE
      write(screen,510)
  510 format(1x,2x,'Weight of Equalizer/Spider C3')
      if (clmp3 ne 9999 99)write(screen,311)clmp3
c     POINT P1
      i=1
      write(screen,520)i,p1x,p1d,p1z
  520 format('+',50x,'Point P',i1,3(1x,f7 2))
c LINE 27
c     LENGTH OF SEGMENT 4
      write(screen,610)
  610 format(1x,2x,'Length of Segment 4',8x,'S4')
      if (scop4 ne 9999 99)write(screen,311)scop4
```

```
c     POINT P2
      i=2
      write(screen,520)i,p2x,p2d,p2z
c     PROJECTED EXCURSION
      write(screen,630)
  630 format('+',85x,'Projected Excursion')
      obmag=sqrt(xbuoy*xbuoy+zbuoy*zbuoy)
      xxproj=xx8
      if (ileg eq 1) xxproj=obmag
      if (xxproj ne 9999.99)write(screen,331)xxproj
c LINE 28
c     LINEAR WEIGHT OF SEGMENT 4
      write(screen,710)
  710 format(1x,2x,'Linear Weight of Segment 4 W4')
      if (wg14 ne 9999.99)write(screen,311)wg14
c     POINT P3
      i=3
      write(screen,520)i,p3x,p3d,p3z
c LINE 29
c     ANCHOR A
      write(screen,820)
  820 format(1x,50x,'Anchor A')
      if (oda ne 9999.99)write(screen,821)oxla,oda,ozla
  821 format('+',58x,3(1x,f7.2))
c     TRUE EXCURSION
      write(screen,830)
  830 format('+',85x,'True Excursion')
      write(screen,331)obmag
c LINE 30
c     ANCHOR B
      write(screen,920)
  920 format(1x,50x,'Anchor B')
      if (odb ne 9999.99)write(screen,821)oxlb,odb,ozlb
c LINE 31
c     ORIGIN
      write(screen,1020)odo
 1020 format(1x,50x,'Origin',6x,'0.00',1x,f7.2,4x,'0.00')
      return
      end
*
```

```
e: sys final/12for/grou11 for##
      subroutine grou11
c
c PRINT VALUES FOR THE UNKNOWN INPUTS
c
      implicit integer*2 (*)
c
c COMMON BLOCK DECLARATIONS
c
c     LUNITS
c
      integer*2       screen,keybd,lul,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /lunits/ screen,keybd,lul,lu2,niv99,siz99,ncp1,
     &    pref1,dum1,ext1,ext2
c
c     VARIN
c
      integer*2 illeg,list
      integer*4 nnca,nncb
      real param(40)
      real angla,anglb,
     &    scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
     &    scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &    scop3a,scop3b,wgt3a,wgt3b,
     &    slip,fric1,                clmp3,
     &    scop4,        wgt4,
     &    anksep,
     &    p1x,p1z,p1d,
     &    p2x,p2z,p2d,
     &    p3x,p3z,p3d,
     &    hload,hdir,
     &    rbuoy,xbuoy,zbuoy,
     &    deptho,pdir
      common /varin/ illeg,list,
     &    nnca,nncb,
     &    angla,anglb,
     &    scop1a,scop1b,wgt1a,wgt1b,clmp1a,clmp1b,
     &    scop2a,scop2b,wgt2a,wgt2b,clmp2a,clmp2b,
     &    scop3a,scop3b,wgt3a,wgt3b,
     &    slip,fric1,                clmp3,
     &    scop4,        wgt4,
     &    anksep,
```

```
     &     p1x ,p1z ,p1d ,
     &     p2x ,p2z ,p2d ,
     &     p3x ,p3z ,p3d ,
     &     hload ,hdir ,
     &     rbuoy ,xbuoy ,zbuoy ,
     &     depth0 ,pdir
      equivalence (param(1),angle)
c
c     VAROUT
c
      real ola ,olb ,ol ,
     &     oha ,ohb ,oh ,
     &     ox1a ,ox3a ,ox5a ,ox1b ,ox3b ,ox5b ,ox7 ,ox8 ,
     &     oy1a ,oy3a ,oy5a ,oy1b ,oy3b ,oy5b ,oy7 ,oy8 ,
     &     oz1a ,oz3a ,oz5a ,oz1b ,oz3b ,oz5b ,oz7 ,oz8 ,
     &     oa1a ,oa2a ,oa3a ,oa4a ,oa5a ,oa6a ,
     &     oa1b ,oa2b ,oa3b ,oa4b ,oa5b ,oa6b ,oa7 ,oa8 ,
     &     ov1a ,ov2a ,ov3a ,ov4a ,ov5a ,ov6a ,
     &     ov1b ,ov2b ,ov3b ,ov4b ,ov5b ,ov6b ,ov7 ,ov8 ,
     &     ot1a ,ot2a ,ot3a ,ot4a ,ot5a ,ot6a ,
     &     ot1b ,ot2b ,ot3b ,ot4b ,ot5b ,ot6b ,ot7 ,ot8 ,
     &     odo ,oda ,odb ,
     &     oaf ,oafdir ,oafa ,oadir ,oafb ,obdir ,
     &     oslp ,ocoila ,ocoilb
      integer*2 oisol ,obrnch
      common /varout/ ola ,olb ,ol ,
     &     oha ,ohb ,oh ,
     &     ox1a ,ox3a ,ox5a ,ox1b ,ox3b ,ox5b ,ox7 ,ox8 ,
     &     oy1a ,oy3a ,oy5a ,oy1b ,oy3b ,oy5b ,oy7 ,oy8 ,
     &     oz1a ,oz3a ,oz5a ,oz1b ,oz3b ,oz5b ,oz7 ,oz8 ,
     &     oa1a ,oa2a ,oa3a ,oa4a ,oa5a ,oa6a ,
     &     oa1b ,oa2b ,oa3b ,oa4b ,oa5b ,oa6b ,oa7 ,oa8 ,
     &     ov1a ,ov2a ,ov3a ,ov4a ,ov5a ,ov6a ,
     &     ov1b ,ov2b ,ov3b ,ov4b ,ov5b ,ov6b ,ov7 ,ov8 ,
     &     ot1a ,ot2a ,ot3a ,ot4a ,ot5a ,ot6a ,
     &     ot1b ,ot2b ,ot3b ,ot4b ,ot5b ,ot6b ,ot7 ,ot8 ,
     &     odo ,oda ,odb ,
     &     oaf ,oafdir ,oafa ,oadir ,oafb ,obdir ,
     &     oslp ,ocoila ,ocoilb ,
     &     oisol ,obrnch
c
c     UNKNOW
c
```

```
      integer*2        nunk,unk1,unk2,unk3,unk4,unk5
      common /unknow/ nunk,unk1,unk2,unk3,unk4,unk5
c
c     GCB
c
      integer*2 gbuff(24),lugraf,luprfl,ludbug
      common /gcb/ gbuff ,lugraf,luprfl,ludbug
c
c LOCAL VARIABLES
c
      integer*2 iiext,ival,unkixt(40),unkval(40)
      integer*1 piext(26,18)
      integer*2 three,five
c
c DATA INITIALIZATION
c
      data three,five/3,5/

      data unkval/
     & 41,41,41,41,41,41,41,41,41,41,
     & 41,41,34,35,36,
     & 41, 1, 3, 5, 7, 9,11,13,15,17,
     & 41, 2, 4, 6, 8,10,12,14,16,18,
     & 19,20,21,22,23/
      data unkixt/
     & 18,18,18,18,18,18,18,18,18,18,
     & 18,18,15,16,17,
     & 18, 1, 2, 3, 4, 5, 6, 7, 8, 9,
     & 18, 1, 2, 3, 4, 5, 6, 7, 8, 9,
     & 10,11,12,13,14/
      data piext/
     & 'Angle to Bottom           ','Length of Segment 1        ',
     & 'Linear Weight of Segment 1','Weight of Sinker 1         ',
     & 'Length of Segment 2       ','Linear Weight of Segment 2',
     & 'Weight of Sinker 2        ','Length of Segment 3        ',
     & 'Linear Weight of Segment 3',
     & 'Final Slippage SS         ','Friction Coefficient       ',
     & 'Weight of Equalizer/Spider','Length of Segment 4        ',
     & 'Linear Weight of Segment 4',
     & 'Horizontal Load Magnitude ','Horizontal Load Direction ',
     & 'Buoy Excursion            ','INVALID ELEMENT            '/
c
c EXECUTABLE PORTION
```

```
c
c OUTPUT
      call chrsiz(five)
      write(screen,100)
  100 format('+','OUTPUT ')
      call chrsiz(three)
c UNKNOWN INPUTS
      write(screen,105)
  105 format(1x,'UNKNOWN INPUTS ')
      if (oslp  eq  9999.99) goto 210
      nunk=nunk+1
      unk3=unk2
      unk2=unk1
  210 continue
      write(screen,140)
c     FIRST UNKNOWN
      if (oslp  eq  9999.99) goto 310
      write(screen,110) oslp,ptext(1,10) 26
      goto 320
  310 continue
      itext=unktxt(unk1)
      ival =unkval(unk1)
      write(screen,110) param(ival),ptext(1,itext) 26
  320 continue
c     SECOND UNKNOWN
      if (nunk eq 1) goto 900
      itext=unktxt(unk2)
      ival =unkval(unk2)
      write(screen,120) param(ival),ptext(1,itext) 26
c     THIRD UNKNOWN
      if (nunk eq 2) goto 900
      itext=unktxt(unk3)
      ival =unkval(unk3)
      write(screen,130) param(ival),ptext(1,itext) 26
  900 continue
      return
c FORMATS
  110 format(1x,f7.2,' = ',26a1)
  120 format('+',38x,f7.2,' = ',26a1)
  130 format('+',76x,f7.2,' = ',26a1)
  140 format(1x)
      end
*
```

```
e! sys final/12for/grou12 for##
      subroutine grou12
c
c Print computed output values for Graph 1
c
      implicit integer*2 (#)
c
c COMMON BLOCK DECLARATIONS
c
c     LUNITS
c
      integer*2       screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /lunits/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &    pref1,dum1,ext1,ext2
c
c     VAROUT
c
      real ola,olb,ol,
     &    oha,ohb,oh,
     &    ox1a,ox3a,ox5a,ox1b,ox3b,ox5b,ox7,ox8,
     &    oy1a,oy3a,oy5a,oy1b,oy3b,oy5b,oy7,oy8,
     &    oz1a,oz3a,oz5a,oz1b,oz3b,oz5b,oz7,oz8,
     &    oa1a,oa2a,oa3a,oa4a,oa5a,oa6a,
     &    oa1b,oa2b,oa3b,oa4b,oa5b,oa6b,oa7,oa8,
     &    ov1a,ov2a,ov3a,ov4a,ov5a,ov6a,
     &    ov1b,ov2b,ov3b,ov4b,ov5b,ov6b,ov7,ov8,
     &    ot1a,ot2a,ot3a,ot4a,ot5a,ot6a,
     &    ot1b,ot2b,ot3b,ot4b,ot5b,ot6b,ot7,ot8,
     &    odo,oda,odb,
     &    oaf,oafdir,oafa,oadir,oafb,obdir,
     &    oslp,ocoila,ocoilb
      integer*2 oisol,obrnch
      common /varout/ ola,olb,ol,
     &    oha,ohb,oh,
     &    ox1a,ox3a,ox5a,ox1b,ox3b,ox5b,ox7,ox8,
     &    oy1a,oy3a,oy5a,oy1b,oy3b,oy5b,oy7,oy8,
     &    oz1a,oz3a,oz5a,oz1b,oz3b,oz5b,oz7,oz8,
     &    oa1a,oa2a,oa3a,oa4a,oa5a,oa6a,
     &    oa1b,oa2b,oa3b,oa4b,oa5b,oa6b,oa7,oa8,
     &    ov1a,ov2a,ov3a,ov4a,ov5a,ov6a,
     &    ov1b,ov2b,ov3b,ov4b,ov5b,ov6b,ov7,ov8,
     &    ot1a,ot2a,ot3a,ot4a,ot5a,ot6a,
```

```
     &     o11b,o12b,o13b,o14b,o15b,o16b,o17,o18,
     &     odo,oda,odb,
     &     oaf,oafdir,oafa,oadir,oafb,obdir,
     &     oslp,oco11a,oco11b,
     &     o1sol,obrnch
c
c LOCAL VARIABLES
c
      integer*1 juntxt(8,2),tentxt(10,3)
      integer*2 rownm(11),junc,tenleg
      data rownm/'HAVAC L H X Y Z A V T '/
      data juntxt/'on floor','elevated'/
      data tentxt/'both legs ','Leg A only','Leg B only'/
c
c EXECUTABLE PORTION
c
c PRINT LEGEND OF ROWS, AND COLUMN HEADERS
      write(screen,1000)
 1000 format(/
     &1x,'HA - Floor Horizontal Angle      VA - Floor Vertical Angle',2x,
     &   'C  - Chain Coiled on Bottom   L  - Length Along Bottom'/
     &1x,'H  - Horizontal Force           X  - X Coordinate',10x,
     &   'Y  - Y Coordinate             Z  - Z Coordinate'/
     &1x,'A  - Catenary Horizontal Angle  V  - Vertical Force',8x,
     &   'T  - Tension'//
     &1x,'    --1A--  --2A--  --3A--  --4A--  --5A--  --6A--',
     &   '  --1B--  --2B--  --3B--  --4B--  --5B--  --6B--',
     &   '  ---7---  ---8---')
c PRINT HA - FLOOR HORIZONTAL ANGLE
      write(screen,1011) rownm(1),oadir
      if (obdir ne 9999 99) write(screen,1070) obdir
c PRINT VA - FLOOR VERTICAL ANGLE
      write(screen,1011) rownm(2),oafa
      if (oafb ne 9999 99) write(screen,1070) oafb
c PRINT C - CHAIN COILED ON THE OCEAN FLOOR
      write(screen,1013) rownm(3)
      if (oco11a  ne  9999 99) write(screen,1060) oco11a
      if (oco11b  ne  9999 99) write(screen,1120) oco11b
c PRINT L - LENGTH ALONG THE OCEAN FLOOR
      write(screen,1012) rownm(4),ola
      if (olb  ne 9999 99) write(screen,1070) olb
      if (ol   ne 9999 99) write(screen,1130) ol
c PRINT H - HORIZONTAL FORCE AT THE BUOY
```

```
      write(screen,1012) rownm(5),oha
      if (ohb  ne 9999 99) write(screen,1070) ohb
      if (oh   ne 9999 99) write(screen,1130) oh
c PRINT X - X COORDINATE OF JUNCTION POINT
      write(screen,1150)
      write(screen,1150)
      write(screen,1010) rownm(6),ox1a
      if (ox3a ne 9999 99) write(screen,1030) ox3a
      if (ox5a ne 9999 99) write(screen,1050) ox5a
      if (ox1b ne 9999 99) write(screen,1070) ox1b
      if (ox3b ne 9999 99) write(screen,1090) ox3b
      if (ox5b ne 9999 99) write(screen,1110) ox5b
      if (ox7  ne 9999 99) write(screen,1130) ox7
      if (ox8  ne 9999 99) write(screen,1140) ox8
c PRINT Y - Y COORDINATE OF JUNCTION POINT
      if (ox8  eq  9999 99) write(screen,1150)
      write(screen,1010) rownm(7),oy1a
      if (oy3a ne 9999 99) write(screen,1030) oy3a
      if (oy5a ne 9999 99) write(screen,1050) oy5a
      if (oy1b ne 9999 99) write(screen,1070) oy1b
      if (oy3b ne 9999 99) write(screen,1090) oy3b
      if (oy5b ne 9999 99) write(screen,1110) oy5b
      if (oy7  ne 9999 99) write(screen,1130) oy7
      if (oy8  ne 9999 99) write(screen,1140) oy8
c PRINT Z - Z COORDINATE OF JUNCTION POINT
      if (oy8  eq  9999 99) write(screen,1150)
      write(screen,1010) rownm(8),oz1a
      if (oz3a ne 9999 99) write(screen,1030) oz3a
      if (oz5a ne 9999 99) write(screen,1050) oz5a
      if (oz1b ne 9999 99) write(screen,1070) oz1b
      if (oz3b ne 9999 99) write(screen,1090) oz3b
      if (oz5b ne 9999 99) write(screen,1110) oz5b
      if (oz7  ne 9999 99) write(screen,1130) oz7
      if (oz8  ne 9999 99) write(screen,1140) oz8
c PRINT A - ANGLE TO THE HORIZONTAL
      if (oz8  eq  9999 99) write(screen,1150)
      write(screen,1150)
      write(screen,1010) rownm(9),oa1a
      if (oa2a ne 9999 99) write(screen,1020) oa2a
      if (oa3a ne 9999 99) write(screen,1030) oa3a
      if (oa4a ne 9999 99) write(screen,1040) oa4a
      if (oa5a ne 9999 99) write(screen,1050) oa5a
      if (oa6a ne 9999 99) write(screen,1060) oa6a
```

```
      if (oa1b ne 9999 99) write(screen,1070) oa1b
      if (oa2b ne 9999 99) write(screen,1080) oa2b
      if (oa3b ne 9999 99) write(screen,1090) oa3b
      if (oa4b ne 9999 99) write(screen,1100) oa4b
      if (oa5b ne 9999 99) write(screen,1110) oa5b
      if (oa6b ne 9999 99) write(screen,1120) oa6b
      if (oa7  ne 9999 99) write(screen,1130) oa7
      if (oa8  ne 9999 99) write(screen,1140) oa8
c PRINT V - VERTICAL FORCE
      if (oa8  eq  9999 99) write(screen,1150)
      write(screen,1010) rownm(10),ov1a
      if (ov2a ne 9999 99) write(screen,1020) ov2a
      if (ov3a ne 9999 99) write(screen,1030) ov3a
      if (ov4a ne 9999 99) write(screen,1040) ov4a
      if (ov5a ne 9999 99) write(screen,1050) ov5a
      if (ov6a ne 9999 99) write(screen,1060) ov6a
      if (ov1b ne 9999 99) write(screen,1070) ov1b
      if (ov2b ne 9999 99) write(screen,1080) ov2b
      if (ov3b ne 9999 99) write(screen,1090) ov3b
      if (ov4b ne 9999 99) write(screen,1100) ov4b
      if (ov5b ne 9999 99) write(screen,1110) ov5b
      if (ov6b ne 9999 99) write(screen,1120) ov6b
      if (ov7  ne 9999 99) write(screen,1130) ov7
      if (ov8  ne 9999 99) write(screen,1140) ov8
c PRINT T - TENSION
      if (ov8  eq  9999 99) write(screen,1150)
      write(screen,1010) rownm(11),ot1a
      if (ot2a ne 9999 99) write(screen,1020) ot2a
      if (ot3a ne 9999 99) write(screen,1030) ot3a
      if (ot4a ne 9999 99) write(screen,1040) ot4a
      if (ot5a ne 9999 99) write(screen,1050) ot5a
      if (ot6a ne 9999 99) write(screen,1060) ot6a
      if (ot1b ne 9999 99) write(screen,1070) ot1b
      if (ot2b ne 9999 99) write(screen,1080) ot2b
      if (ot3b ne 9999 99) write(screen,1090) ot3b
      if (ot4b ne 9999 99) write(screen,1100) ot4b
      if (ot5b ne 9999 99) write(screen,1110) ot5b
      if (ot6b ne 9999 99) write(screen,1120) ot6b
      if (ot7  ne 9999 99) write(screen,1130) ot7
      if (ot8  ne 9999 99) write(screen,1140) ot8
c PRINT SOLUTION TYPE USED FOR COMPOUND LEG
      if (ot8  eq  9999 99) write(screen,1150)
      junc = (ot+sol+1) / 2
```

```
      tenleg = obrnch + 1
      if (ox1b .ne. 9999.99) write(screen,1160) juntxt(1,junc) 8,
     & tentxt(1,tenleg) 10
      return
 1010 format('+',a1,f8.2)
 1011 format(1x,a2,f7.2)
 1012 format(1x,a1,f8.2)
 1013 format(1x,a1)
 1020 format('+',9x,f8.2)
 1030 format('+',17x,f8.2)
 1040 format('+',25x,f8.2)
 1050 format('+',33x,f8.2)
 1060 format('+',41x,f8.2)
 1070 format('+',49x,f8.2)
 1080 format('+',57x,f8.2)
 1090 format('+',65x,f8.2)
 1100 format('+',73x,f8.2)
 1110 format('+',81x,f8.2)
 1120 format('+',89x,f8.2)
 1130 format('+',98x,f8.2)
 1140 format('+',106x,f8.2)
 1150 format(1x)
 1160 format(1x,'Solution Type      Junction ',8a1,',   Tension on ',
     & 10a1)
      end
*
```

```
e! sys final/12for/graphs for##
      subroutine graphs(isw)
      implicit integer*2 (#)
c
c Produce annotated symbolic depiction of compound leg, tables of
c parameters input to define it, and tables of computed forces,
c coordinates, angles and tensions
c
c
c PARAMETER
c
      integer*2 isw
c
c COMMON BLOCK DECLARATIONS
c
c
c     TITLES
c
      integer*1       ctitle(114)
      common /titles/ ctitle
c
c     DATIME
c
      integer*1       cdatim(16)
      common /datime/ cdatim
c
c     VARIN
c
      integer*1      cvarin(172)
      common /varin/ cvarin
c
c     VAROUT
c
      integer*1       cvaro1(240),cvaro2(100)
      common /varout/ cvaro1,     cvaro2
c
c     VARG
c
      integer*1     cvarg(240)
      common /varg/ cvarg
c
c     UNKNOW
c
```

```
      integer*1       cunkno(12)
      common /unknow/ cunkno
c
c     GROPT
c
      integer*1       cgropt(44)
      common /gropt/ cgropt
c
c     GRP2CN
c
      integer*1       cgrp21(218),cgrp22(82)
      common /grp2cn/ cgrp21,     cgrp22
c
c LOCAL
c
      integer*2 funkey
c
c EXECUTABLE PORTION
c
c PRINT TABLES OF INPUT PARAMETERS, COMPUTED VALUES AND STICK FIGURE
      call RWCOM1(1)
      isw=1
      call stick
50    call flush
      call readfk(funkey)
*     if(funkey NE -32768) go to 100
      CALL ERASE
*     WRITE(6,9000)
9000  FORMAT(//////////////,' ',15X,'PROCESSING AT THIS POINT MAY TAKE '
     &,'45 SECONDS   PLEASE BE PATIENT!')
      GO TO 300
100   if(funkey eq 16384) go to 200
      go to 50
200   isw=2
300   continue
      call RWCOM1(2)
      return
      end
*
```

```
et sys final/t2for/stick for##
      SUBROUTINE STICK
C
C THIS ROUTINE WILL PUT OUT THE STICK FIGURE ON THE FIRST GRAPHIC SCREEN
C
      IMPLICIT INTEGER*2 (A-Z,$)
      INTEGER*2 ISW
      INTEGER MINUS, PLUS
      REAL THETA,SCALE
c
c     GCB
c
      integer*2 gbuff(24),lugraf,luptfl,ludbug
      common /gcb/ gbuff,lugraf ,luptfl,ludbug
      DATA MINUS/'-'/, PLUS/'+'/
c
c
c
      call gfinit
      CALL VIEWPT(-1150,32766,6984,19316)
      CALL WINDOW(-1200,32767,6900,19400)
C
C INITIALIZATION DONE, SET OUT FIGURE
C
      ISW=0
      THETA = 0 0
      SCALE = 75 0
C
C DRAW THE BOUNDARY LINES
C
      CALL MOVETO(-1149,6985)
      CALL DRAWTO(32765,6985)
      CALL DRAWTO(32765,19315)
      CALL DRAWTO(-1149,19315)
      CALL DRAWTO(-1149,6985)
C
C SET THE LOWER LEFT ANCHOR AND START DRAWING FIGURES
C
      CALL MOVETO(400,10500)
      CALL ANCHOR(THETA,SCALE,ISW)
      CALL DRAW(8000,0)
      CALL SINKER(THETA,SCALE,ISW)
      CALL DRAW(8000,0)
```

244

```
      CALL SINKER(THETA,SCALE,ISW)
      CALL DRAW(8000,0)
      CALL DASH(0,4600)
      CALL DRAW(-8000,0)
      CALL SINKER(THETA,SCALE,ISW)
      CALL DRAW(-8000,0)
      CALL SINKER(THETA,SCALE,ISW)
      CALL DRAW(-8000,0)
      CALL ANCHOR(THETA,SCALE,ISW)
      CALL MOVE(24000,-2300)
      CALL ELIZER(THETA,SCALE,ISW)
      CALL DRAW(6500,0)
      CALL BUOY(THETA,SCALE,ISW)
C
C END OF PICTURE, NOW TITLE AND BE DONE
C
      CALL MOVETO(-1000,7200)
      WRITE(LUGRAF,1000)
1000  FORMAT(' ',20X,'BRANCH B')
      CALL MOVETO(-1000,10600)
      WRITE(LUGRAF,1001)
1001  FORMAT(' ',3X,'1',11X,'2 3',11X,'4 5',11X,'6')
      CALL MOVETO(-1000,15200)
      WRITE(LUGRAF,1001)
      CALL MOVETO(-1000,14150)
      WRITE(LUGRAF,1002) MINUS
1002  FORMAT(' ',6X,'S1 W1     C1  S2 W2     C2  S3',A1,'S5 W3')
      CALL MOVETO(-1000,9550)
      WRITE(LUGRAF,1002) PLUS
      CALL MOVETO(-1000,18500)
      WRITE(LUGRAF,1003)
1003  FORMAT(' ',20X,'BRANCH A')
      CALL MOVETO(25900,18500)
      WRITE(LUGRAF,1004)
1004  FORMAT(' ','COMMON')
      CALL MOVETO(25900,17600)
      WRITE(LUGRAF,1007)
1007  FORMAT(' ','SECTION')
      CALL MOVETO(24800,12950)
      WRITE(LUGRAF,1005)
1005  FORMAT(' ','7',8X,'8')
      CALL MOVETO(24800,12050)
      WRITE(LUGRAF,1006)
```

```
1006   FORMAT(' ','C3',2X,'S4 W4')
       RETURN
       END
*
```

246

```
e! sys final/i2for/sinker for##
      SUBROUTINE SINKER(THETA,SCALE,ISW)
C
C THIS ROUTINE WILL DRAW AN SINKER SYMBOL ON THE SCREEN AT THE POINT X,Y
C THE ORIENTATION OF THE SYMBOL WILL BE ABOUT THE ANGLE THETA (MEASURED CLOCKWISE)
C THE SCALE FACTOR WILL EITHER SHRINK OR EXAGGERATE THE SYMBOL WITH 1 0 BEING THE NORM
AL
C THE SYMBOL IS CENTERED ON THE POINT X,Y
C
      IMPLICIT REAL (A-Z)
      INTEGER*2 I,IX,IY,IXSV,IYSV,ISW,JSW
      DIMENSION XPNT(4),YPNT(4)
      DIMENSION XPOINT(4), YPOINT(4)
      COMMON /GROPT/WXL,WXU,WYL,WYU
      DATA XPOINT/ 0 0, 2 5, -5 0, 2 5/
      DATA YPOINT/ 3 5,-7 0, 0 0, 7 0/
      DATA DELVX/64000 /, DELVY/48000 /
     DATA XPNT/0 0,280 5,-561 0,280 5/
      DATA YPNT/421 0,-842 0,0 0,842 0/
C
C CHECK IS THE SCALE VALUE LESS THAN ZERO, YES RETURN
C
      IF (SCALE  LE  0 0) GO TO 1000
C
C SAVE THE ORIGINAL VALUES OF THE INPUTS AND DRAW SYMBOL
C
      JSW=ISW+1
      XSC=SCALE
      IF(JSW EQ 2) XSC=1 0
      CT=1 0
      ST=0 0
      IF (THETA  EQ  0 0) GO TO 100
      CT = COS(THETA)
      ST = SIN(THETA)
  100 CONTINUE
     DO 200 I=1,4
         XP=XPOINT(I)
         YP=YPOINT(I)
         IF(JSW EQ 1) GO TO 125
             DELWX=WXU-WXL
             DELWY=WYU-WYL
             XP=(XPNT(I)*DELWX)/DELVX
             YP=(YPNT(I)*DELWY)/DELVY
```

```
  125      XF=(CT*XP + ST*YP) * XSC
           YF=(-ST*XP + CT*YP) * XSC
           IX=XF
           IY=YF
           IF (I GT 1) GO TO 150
  110      CALL MOVE(IX,IY)
  130      IXSV = IX
           IYSV = IY
           GO TO 200
  150      CALL DRAW(IX,IY)
200      CONTINUE
         CALL MOVE(-IXSV,-IYSV)
1000     RETURN
         END
*
```

```
e! sys final/t2for/anchor for$$
      SUBROUTINE ANCHOR(THETA,SCALE,ISW)
C
C THIS ROUTINE WILL DRAW AN ANCHOR SYMBOL (X) ON THE SCREEN AT THE POINT X,Y
C THE ORIENTATION OF THE ANCHOR WILL BE ABOUT THE ANGLE THETA (MEASURED CLOCKWISE)
C THE SCALE FACTOR WILL EITHER SHRINK OR EXAGGERATE THE ANCHOR WITH 1 0 BEING THE NORM
AL
C THE SYMBOL IS CENTERED ON THE POINT X,Y
C
      IMPLICIT REAL (A-Z)
      INTEGER*2 I,IX,IY,ISW,JSW
      DIMENSION XPOINT(4), YPOINT(4)
      DIMENSION XPNT(4),YPNT(4)
      COMMON /CROPT/WXL,WXU,WYL,WYU
      DATA DELVX/64000 /, DELVY/48000 /
      DATA XPNT/280 5,280 5,-280 5,-280 5/
      DATA YPNT/421 0,-421 0,-421 0,421 0/
      DATA XPOINT/ 2 5, 2 5, -2 5, -2 5/
      DATA YPOINT/ 3 5,-3 5, -3 5, 3 5/
C
C CHECK IS THE SCALE VALUE LESS THAN ZERO, YES RETURN
C
      IF (SCALE  LE  0 0) GO TO 500
C
C SAVE THE ORIGINAL VALUES OF THE INPUTS AND DRAW SYMBOL
C
      JSW=ISW+1
      XSC=SCALE
      IF(JSW EQ 2) XSC=1 0
      CT=1 0
      ST=0 0
      IF (THETA  EQ  0 0) GO TO 100
      CT = COS(THETA)
      ST = SIN(THETA)
  100 CONTINUE
      DO 200 I=1,4
         XP=XPOINT(I)
         YP=YPOINT(I)
         IF(JSW EQ 1) GO TO 125
            DELWX=WXU-WXL
            DELWY=WYU-WYL
            XP=(XPNT(I)*DELWX)/DELVX
            YP=(YPNT(I)*DELWY)/DELVY
```

```
  125      XF=(CT*XP + ST*YP) * XSC
           YF=(-ST*XP + CT*YP) * XSC
           IX=XF
           IY=YF
           CALL DRAW(IX,IY)
           CALL MOVE(-IX,-IY)
  200   CONTINUE
  500   RETURN
        END
*
```

```
e! sys final/12for/elizer for##
      SUBROUTINE ELIZER(THETA,SCALE,ISW)
C
C THIS ROUTINE WILL DRAW AN ELIZER SYMBOL ON THE SCREEN AT THE POINT X,Y
C THE ORIENTATION OF THE SYMBOL WILL BE ABOUT THE ANGLE THETA (MEASURED CLOCKWISE)
C THE SCALE FACTOR WILL EITHER SHRINK OR EXAGGERATE THE SYMBOL WITH 1 0 BEING THE NORM
AL
C THE SYMBOL IS CENTERED ON THE POINT X,Y
C
      IMPLICIT REAL (A-Z)
      INTEGER*2 I,IX,IY,IXSV,IYSV,ISW,JSW
      DIMENSION XPOINT(4), YPOINT(4)
      DIMENSION XPNT(4),YPNT(4)
      COMMON /GROPT/WXL,WXU,WYL,WYU
      DATA DELVX/64000 /, DELVY/48000 /
      DATA XPNT/561 0,-561 0,0 0,561 0/
      DATA YPNT/0 0,-421 0,842 0,-421 0/
      DATA XPOINT/ 5 0, -5 0, 0 0, 5 0/
      DATA YPOINT/ 0 0,-3 5, 7 0, -3 5/
C
C CHECK IS THE SCALE VALUE LESS THAN ZERO, YES RETURN
C
      IF (SCALE  LE  0 0) GO TO 1000
C
C SAVE THE ORIGINAL VALUES OF THE INPUTS AND DRAW SYMBOL
C
      JSW=ISW+1
      XSC=SCALE
      IF(JSW EQ 2) XSC=1 0
      CT=1 0
      ST=0 0
      IF (THETA  EQ  0 0) GO TO 100
      CT = COS(THETA)
      ST = SIN(THETA)
  100 CONTINUE
      DO 200 I=1,4
          XP=XPOINT(I)
          YP=YPOINT(I)
          IF(JSW EQ 1) GO TO 125
              DELWX=WXU-WXL
              DELWY=WYU-WYL
              XP=(XPNT(I)*DELWX)/DELVX
              YP=(YPNT(I)*DELWY)/DELVY
```

```
  125      XF=(CT*XP + ST*YP) * XSC
           YF=(-ST*XP + CT*YP) * XSC
           IX=XF
           IY=YF
           IF (I NE 1) GO TO 150
  110      CALL MOVE(IX,IY)
  130      IXSV = IX
           IYSV = IY
           GO TO 200
  150      CALL DRAW(IX,IY)
200      CONTINUE
210      CALL MOVE(-IXSV,-IYSV)
1000     RETURN
         END
*
```

```
e! sys final/:2for/buoy for$$
      SUBROUTINE BUOY(THETA,SCALE,ISW)
C
C THIS ROUTINE WILL DRAW AN BUOY SYMBOL ON THE SCREEN AT THE POINT X,Y
C THE ORIENTATION OF THE SYMBOL WILL BE ABOUT THE ANGLE THETA (MEASURED CLOCKWISE)
C THE SCALE FACTOR WILL EITHER SHRINK OR EXAGGERATE THE SYMBOL WITH 1 0 BEING THE NORM
AL
C THE SYMBOL IS CENTERED ON THE POINT X,Y
C
       IMPLICIT REAL (A-Z)
       INTEGER*2 I,IX,IY,IXSV,IYSV,ISW,JSW
       DIMENSION XPOINT(5), YPOINT(5)
       DIMENSION XPNT(5),YPNT(5)
      COMMON /CROPT/WXL,WXU,WYL,WYU
       DATA XPOINT/2 5, 0 0, -5 0, 0 0, 5 0/
       DATA YPOINT/ 3 5,-7 0, 0 0,7 0,0 0/
      DATA DELVX/64000 /, DELVY/48000 /
       DATA XPNT/280 5,0 0,-561 0,0 0,561 0/
       DATA YPNT/421 0,-842 0,0 0,842 0,0 0/
C
C CHECK IS THE SCALE VALUE LESS THAN ZERO, YES RETURN
C
       IF (SCALE  LE  0 0) GO TO 1000
C
C SAVE THE ORIGINAL VALUES OF THE INPUTS AND DRAW SYMBOL
C
       JSW=ISW+1
       XSC=SCALE
      IF(JSW EQ 2) XSC=1 0
       CT=1 0
       ST=0 0
       IF (THETA  EQ  0 0) GO TO 100
       CT = COS(THETA)
       ST = SIN(THETA)
  100  CONTINUE
      DO 200 I=1,5
          XP=XPOINT(I)
          YP=YPOINT(I)
          IF(JSW EQ 1) GO TO 125
              DELWX=WXU-WXL
              DELWY=WYU-WYL
              XP=(XPNT(I)*DELWX)/DELVX
              YP=(YPNT(I)*DELWY)/DELVY
```

```
  125      XF=(CT*XP + ST*YP) * XSC
           YF=(-ST*XP + CT*YP) * XSC
           IX=XF
           IY=YF
           IF (I NE 1) GO TO 150
  110      CALL MOVE(IX,IY)
           IXSV = IX
           IYSV = IY
           GO TO 200
  150      CALL DRAW(IX,IY)
200      CONTINUE
210      CALL MOVE(-IXSV,-IYSV)
1000     RETURN
         END
*
```

```
e! sys final/f2for/elvpnt for##
      subroutine ELVPNT(iov,ifil,isw)
*****************************************************************************
      implicit integer*2 (*)
      implicit double precision (a-z)

      integer*2 iov,ifil,isw

      integer*2 screen,keybd,lul,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),duml,ext1(4),ext2(4),ext3(4)
      common /LUNITS/ screen,keybd,lul,lu2,niv99,siz99,ncpl,
     &    prefl,duml,ext1,ext2,ext3

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
```

```
      equivalence (z(51),col1),(z(52),slp),(z(53),frct),(z(54),c3),
     &   (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &   (z(59),tana7),(z(60),tana8),(z(61),l),
     &   (z(62),hl),(z(63),phih),
     &   (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      integer*2 nc(2)
      equivalence (nca,nc)
      double precision tx(2)
      equivalence (ta,tx)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &   izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*2 idate(5),ihour,imin,isec
      common /DATIME/ idate,ihour,imin,isec

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*1 cvarol(240),cvaro2(100)
      common /VAROUT/ cvarol,cvaro2

      double precision ddum1(13),
     &   ga11,ga12,ga21,ga22,ga31,ga32,
     &   gb11,gb12,gb21,gb22,gb31,gb32,
     &   g1,g2,ddum2(3)
      common /VARC/ ddum1
     &   ga11,ga12,ga21,ga22,ga31,ga32,
     &   gb11,gb12,gb21,gb22,gb31,gb32,
     &   g1,g2,ddum2
```

```
      double precision gcff(12)
      equivalence (gall,gcff)

      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      double precision cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1
      real xout,yout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint(5)
      common /VELVPT/cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1,xout,yout,
     &  isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint

      integer*2 nbr,ibc,icurv,ibent,ip,
     &  iend,ivert,ils,ig,igtyp,i,j,n,igc
      double precision xf(2,2),yf(2,2)

      integer*1 ans(1),yes
      data yes/'Y'/
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      if (ifil eq 1) goto 30
      write(screen,*) 'Do you want to save output for elevation views?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 20
      isw=1
      goto 9000
   20 continue
      write(screen,*) 'Enter segment increment length (feet) '
      read(keybd,*) seglen
      call RWCOM1(1)
      goto 100

   30 continue
      if (isw eq 1) goto 9000
```

```
      if (iov eq 1) goto 50
      call ADDEXT(ifile,31,ext3)
      call file(i2file,lut,3)
      goto 55
   50 continue
      call ADDEXT(ofile,31,ext3)
      call file(o2file,lut,3)
   55 continue
      igtyp=2
      write(lut,4) igtyp
      write(lut,1) title
      write(lut,2) idate
      write(lut,3) ihour,imin,isec
      xout=xmax
      yout=do
      if (ileg eq 1) yout=yo
      write(lut,7) xout,yout
      xout=xmin
      yout=ymin
      write(lut,7) xout,yout
      write(lut,5) ncomp
  100 continue

      phip=phih
      if (ileg eq 1) goto 120
      nbr=2
      xk=halfd*dsin(phip)
      yk=half*delyk
      goto 150
  120 continue
      nbr=1
      xk=zero
      yk=zero
  150 continue
      ibc=3-ibranch
      xmin=zero
      ymin=zero
      ncomp=0
      igc=0

      do 5000 ib=1,nbr
           icurv=0
           if (ileg eq 1  or  isol eq 4) icurv=1
```

```
      if (isol eq 3  and  x(ibc) ne zero  and  ib eq ibrnch)
     &   icurv=1
      ibent=1
      if (ileg eq 1  or  isol ne 3) ibent=0
      if (x(ibc) eq zero  or  ib eq ibrnch) ibent=0
      ncx=nc(ib)
      ioff=25*(ib-1)
      ip=ioff+25
      ix=z(ip-1)
      phix=z(ip)
      if (ileg eq 1) phix=phih
      snafx=dcos(phix-phif)*snaf
      csafx=one/SECNT(snafx)
      snafx=snafx*csafx
      cosdp=dcos(phix-phip)

      sf=zero
      do 210 ic=1,ncx
           is=ioff+3*ic
           sf=sf+z(is)
210        continue
      sf=sf*1.0d-1
      xf(1,ib)=xk-sf*csafx*cosdp
      yf(1,ib)=yk-sf*snafx
      xf(2,ib)=xk+(ix+sf)*csafx*cosdp
      yf(2,ib)=yk+(ix+sf)*snafx
      xmin=dmin1(xmin,xf(1,ib))
      do 250 i=1,2
           ymin=dmin1(ymin,yf(i,ib))
250        continue

      if (ifil eq 0) goto 300
      igc=igc+1
      write(lu1,6) npoint(igc)
300   continue
      npt=0
      xg=zero
      yg=zero
      isym=5
      call WELVPT(ifil)

      xsum0=zero
      ysum0=zero
```

```
      ssum0=zero
      iend=0
      iver1=0
      do 4000 ic=1,ncx
            ix=ioff+12+ic
            iy=ix+3
            is=ioff+3*ic
            xsum1=xsum0+z(ix)
            ysum1=ysum0+z(iy)
            ssum1=ssum0+z(is)
            isym=0
            if (iben1 eq 1) goto 2000

            if (ix le ssum0) goto 1500
            if (ix ge ssum1) goto 1200
            ils=1
            xg=ix
            if (icurv eq 1) goto 1120
            iend=1
1120        continue
            goto 1300
1200        continue
            ils=2
            xg=ssum1
            if (ic eq ncx) goto 1250
            if (z(ix+1) eq zero) goto 1250
            call SYMSNK
            goto 1300
1250        continue
            iend=1
1300        continue
            if (iend ne 1) goto 1400
            if (ib ne 2) goto 1400
            if (isol ne 2 and isol ne 3) goto 1320
            isym=3
            goto 1400
1320        continue
            if (ileg eq 3) goto 1330
            isym=4
            goto 1400
1330        continue
            isym=2
1400        continue
```

```
                xg=xg*csafx
                yg=xg*tnafx
                call WELVPT(ifil)
                if (iend eq 1) goto 4100
                if (ils eq 2) goto 3000
1500            continue
                ig=6*(ib-1)+2*(ic-1)+1
                gx1=gcff(ig)
                gx2=gcff(ig+1)
                call ELVCAT(0,ifil)
                goto 3000

2000            continue
                if (ic ne 1) goto 2100
                if (ibrnch ne 1) goto 2020
                lh=LENH(ib,ncb,zb)
                goto 2100
2020            continue
                lh=LENH(ia,nca,za)
2100            continue
                ssum2=zero
                do 2150 i=1,ncx
                     j=ncx+1-i
                     if (j le ic) goto 2150
                     j=ioff+15+j
                     ssum2=ssum2+z(j)
2150                 continue
                if (ivert eq 1) goto 2500
                if (lx lt ssum1) goto 2400
                xg=ssum1*csafx
                yg=xg*tnafx
                if (lx eq ssum1) goto 2220
                call SYMSNK
                goto 2250
2220            continue
                ivert=1
                isym=3
2250            continue
                call WELVPT(ifil)
                goto 3000
2400            continue
                xg=lx*csafx
                yg=xg*tnafx
```

```
                ivert=1
                isym=3
                call WELVPT(ifil)
2500            continue
                if (ssum2 ge lh) goto 3000
                yg=ysum1
                if (ic eq ncx) goto 2520
                call SYMSNK
                goto 2600
2520            continue
                if (ib eq 2) goto 2530
                isym=0
                goto 2600
2530            continue
                if (ileg ne 2) goto 2540
                isym=4
                goto 2600
2540            continue
                isym=2
2600            continue
                call WELVPT(ifil)

3000            continue
                xsum0=xsum1
                ysum0=ysum1
                ssum0=ssum1
                ymin=dmin1(ymin,yk+yg)
4000            continue


4100        continue
            ncomp=ncomp+1
            npoint(ncomp)=npt
            xk= -xk
            yk= -yk
5000        continue

      if (ileg eq 1) goto 6000
      xk= -xk+xg*cosdp
      yk= -yk+yg
      cosdp=dcos(phih-phip)
      if (ifil eq 0) goto 5100
      igc=igc+1
```

```
      write(lul,6) npoint(igc)
5100  continue
      npt=0
      xg=zero
      yg=zero
      isym=0
      call WELVPT(ifil)
      if (l eq zero) goto 5200
      inafx=dcos(phih-phif)*inaf
      csafx=one/SECNT(inafx)
      xg=l*csafx
      yg=xg*inafx
      call WELVPT(ifil)
5200  continue
      gxl=gl
      gx2=g2
      lx=l
      ssum0=zero
      ssuml=s4
      xsum0=zero
      xsuml=x4
      ysuml=y4
      is=55
      ioff=61
      call ELVCAT(1,,ifil)
      ncomp=ncomp+l
      npoint(ncomp)=npt

6000  continue
      xmax=xout

      isym=7
      do 6200 ib=l,nbr
           if (ifil eq 0) goto 6130
           igc=igc+l
           write(lul,6) npoint(igc)
           do 6120 i=l,2
                xout=xf(i,ib)
                yout=yf(i,ib)
                write(lul,8) xout,yout,,isym
6120            continue
6130       continue
           ncomp=ncomp+l
```

```
            npoint(ncomp)=2
 6200       continue
      call close(lu1)

 9000 continue
      return

    1 format(50a1)
    2 format(5a2)
    3 format(i2,' ',i2,' ',i2)
    4 format(i1)
    5 format(i2)
    6 format(i5)
    7 format(f8 2,1x,f8 2)
    8 format(f8 2,1x,f8 2,i2)
    9 format(19a1)
      end
*
```

```
e! sys final/t2for/symsnk for##
      subroutine SYMSNK
*****************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VCLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1
      real xout,yout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint(5)
      common /VELVPT/cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1,xout,yout,
     &  isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      clmp=z(is+2)
      if (clmp ge zero) goto 10
      isym=4
      goto 100
   10 continue
      if (clmp gt zero) goto 20
      isym=1
      goto 100
   20 continue
      isym=2
  100 continue
      return
      end
*
```

```
e! sys final/t2for/welvpt for##
      subroutine WELVPT(ifil)
*****************************************************************************
      implicit integer*2 (#)

      integer*2 ifil

      integer*2 screen,keybd,lul,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),duml,extl(4),ext2(4)
      common /LUNITS/ screen,keybd,lul,lu2,niv99,siz99,ncpl,
     &    prefl,duml,extl,ext2

      double precision cosdp,xk,yk,xg,yg,gxl,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssuml,xsum0,xsuml,ysum0,ysuml
      real xout,yout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint(5)
      common /VELVPT/cosdp,xk,yk,xg,yg,gxl,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssuml,xsum0,xsuml,ysum0,ysuml,xout,yout,
     &  isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      xout=xk+xg*cosdp
      yout=yk+yg
      if (ifil eq 0) goto 100
      write(lul,8) xout,yout,isym
  100 continue
      npt=npt+1
      return

    8 format(f8.2,1x,f8.2,i2)
      end
*
```

```
e: sys final/t2for/elvcat for;#
      subroutine ELVCAT(iris,ifil)
*************************************************************************
      implicit integer*2 (i)
      implicit double precision (a-z)

      integer*2 iris,ifil

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,iz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,iz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1
      real xout,yout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint(5)
      common /VELVPT/cosdp,xk,yk,xg,yg,gx1,gx2,lx,seglen,xmin,ymin,
     &  xmax,ssum0,ssum1,xsum0,xsum1,ysum0,ysum1,xout,yout,
     &  isym,npt,ib,ncx,ioff,ic,ix,iy,is,ncomp,npoint

      integer*2 nsegs,i
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
      lngx1=dlog(gx1)
      scop=ssum1-dmax1(lx,ssum0)
      nsegs=(scop*0 999999d0)/seglen+1
      sgl=scop/nsegs
      wx=z(is+1)
      hx=z(ioff+1)
      hw=hx/wx
      wh=wx/hx
      isym=0
      xgoff=xg
      xxg=xg-xgoff
      do 1600 i=1,nsegs
           if (i ne 1) goto 1510
```

```
      temp= -hw*lngx1
      if (temp lt xxg  or  temp gt xsum1-xgoff) goto 1510
      ymin=dmin1(ymin,yk+hw+gx2)
1510  continue
      if (ifil eq 0) goto 1700
      if (i eq nsegs) goto 1520
      temp=gx1*dexp(wh*xxg)
      temp=wh*sgl+half*(temp-one/temp)
      temp=temp+SECNT(temp)
      xxg=hw*(dlog(temp)-lngx1)
      xg=xgoff+xxg
      yg=hw*half*(temp+one/temp)+gx2
      goto 1580
1520  continue
      xg=xsum1
      yg=ysum1
      if (ifis ne 1) goto 1550
      isym=3
      goto 1580
1550  continue
      if (ic eq ncx1 goto 1560
      call SYMSNK
      goto 1580
1560  continue
      if (ileg ne 1) goto 1570
      isym=3
      goto 1580
1570  continue
      if (ib ne 2) goto 1580
      if (ileg ne 2) goto 1575
      isym=4
      goto 1580
1575  continue
      isym=2
1580  continue
      call WELVPT(ifil)
1600  continue
    goto 1800

1700 continue
    npt=npt+nsegs
    xg=xsum1
    yg=ysum1
```

```
      xout=xk+xg*cosdp
      yout=yk+yg
 1800 continue

      return
      end
*
```

```
e! sys final/f2for/plnpnt for$$
      subroutine PLNPNT(iov,ifil,isw)
*************************************************************************
      implicit integer*2 (*)
      implicit double precision (a-z)

      integer*2 iov,ifil,isw

      integer*2 screen,keybd,lul,lu2,niv99,siz99,ncpl
      integer*1 prefl(21),duml,ext1(4),ext2(4),ext3(4),ext4(4)
      common /LUNITS/ screen,keybd,lul,lu2,niv99,siz99,ncpl,
     &    prefl,duml,ext1,ext2,ext3,ext4

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
```

```
      equivalence (z(51),coll),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
      integer*2 nc(2)
      equivalence (nca,nc)
      double precision tx(2)
      equivalence (ta,tx)

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*1 title(50),ifile(32),ofile(32)
      common /TITLES/ title,ifile,ofile
      integer*2 i2file(16),o2file(16)
      equivalence (ifile,i2file),(ofile,o2file)

      integer*2 idate(5),ihour,imin,isec
      common /DATIME/ idate,ihour,imin,isec

      integer*1 cvarin(172)
      common /VARIN/ cvarin

      integer*1 cvaro1(240),cvaro2(100)
      common /VAROUT/ cvaro1,cvaro2

      double precision ddum1(13),
     &  ga11,ga12,ga21,ga22,ga31,ga32,
     &  gb11,gb12,gb21,gb22,gb31,gb32,
     &  g1,g2,ddum2(3)
      common /VARG/ ddum1,
     &  ga11,ga12,ga21,ga22,ga31,ga32,
     &  gb11,gb12,gb21,gb22,gb31,gb32,
     &  g1,g2,ddum2
```

```
      double precision gcff(12)
      equivalence (gall,gcff)

      integer*1 cunkno(12)
      common /UNKNOW/ cunkno

      integer*1 cgropt(44)
      common /GROPT/ cgropt

      integer*1 cgrp21(218),cgrp22(82)
      common /GRP2CN/ cgrp21,cgrp22

      double precision cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1
      real xout,zout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint(5)
      common /VELVPT/cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1,xout,zout,
     &  isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint

      integer*2 nbr,ibc,icurv,ibent,ip,
     &  iend,iverf,ils,ig,igtyp,i,j,n,igc

      integer*1 ans(1),yes
      data yes/'Y'/
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
      if (ifil eq 1) goto 30
      write(screen,*) 'Do you want to save output for plan views?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 20
      isw=1
      goto 9000
   20 continue
      call RWCOM1(1)
      goto 100

   30 continue
      if (isw eq 1) goto 9000
      if (iov eq 1) goto 50
      call ADDEXT(ifile,31,ext4)
      call file(i2file,lu1,3)
```

```
      goto 55
   50 continue
      call ADDEXT(ofile,31,ext4)
      call file(o2file,lul,3)
   55 continue
      igtyp=3
      write(lul,4) igtyp
      write(lul,1) title
      write(lul,2) idate
      write(lul,3) ihour,imin,isec
      xout=xmax
      zout=zmax
      write(lul,7) xout,zout
      xout=0.0
      zout=zmin
      write(lul,7) xout,zout
      write(lul,5) ncomp
  100 continue

      if (ileg eq 1) goto 120
      nbr=2
      zk=halfd
      goto 150
  120 continue
      nbr=1
      zk=zero
  150 continue
      xk=zero
      ibc=3-ibrnch
      zmin= -halfd
      zmax= halfd
      ncomp=0
      igc=0

      do 5000 ib=1,nbr
           icurv=0
           if (ileg eq 1  or  isol eq 4) icurv=1
           if (isol eq 3  and  tx(ibc) ne zero  and  ib eq ibrnch)
     &        icurv=1
           ibent=1
           if (ileg eq 1  or  isol ne 3) ibent=0
           if (tx(ibc) eq zero  or  ib eq ibrnch) ibent=0
           ncx=nc(ib)
```

```
      ioff=25*(ib-1)
      ip=ioff+25
      lx=z(ip-1)
      phix=z(ip)
      if (ileg eq 1) phix=phih
      cospx=dcos(phix)
      sinpx=dsin(phix)
      tnafx=dcos(phix-phif)*tnaf
      csafx=one/SECNT(tnafx)
      snafx=tnafx*csafx

      if (ifil eq 0) goto 300
      igc=igc+1
      write(lul,6) npoint(igc)
300   continue
      npt=0
      xxg=zero
      isym=5
      call WPLNPT(ifil)

      xsum0=zero
      ssum0=zero
      iend=0
      ivert=0
      do 4000 ic=1,ncx
           ix=ioff+12+ic
           is=ioff+3*ic
           xsuml=xsum0+z(ix)
           ssuml=ssum0+z(is)
           isym=0
           if (ibent eq 1) goto 2000

           if (lx le ssum0) goto 1500
           if (lx ge ssuml) goto 1200
           ils=1
           xxg=lx
           if (icurv eq 1) goto 1120
           iend=1
1120       continue
           goto 1300
1200       continue
           ils=2
           xxg=ssuml
```

```
                if (ic eq ncx) goto 1250
                if (z(ix+1) eq zero) goto 1250
                call PLNSNK
                goto 1300
1250            continue
                iend=1
1300            continue
                if (iend ne 1) goto 1400
                if (ib ne 2) goto 1400
                if (isol ne 2  and  isol ne 3) goto 1320
                isym=3
                goto 1400
1320            continue
                if (ileg eq 3) goto 1330
                isym=4
                goto 1400
1330            continue
                isym=2
1400            continue
                xxg=xxg*csafx
                call WPLNPT(if,il)
                if (iend eq 1) goto 4100
                if (ils eq 2) goto 3000
1500            continue
                call PLNCAT(0,if,il)
                goto 3000

2000            continue
                if (ic ne 1) goto 2100
                if (ibrnch ne 1) goto 2020
                lh=LENH(ib,ncb,zb)
                goto 2100
2020            continue
                lh=LENH(ia,nca,za)
2100            continue
                ssum2=zero
                do 2150 i=1,ncx
                     j=ncx+1-i
                     if (j le ic) goto 2150
                     j=ioff+15+j
                     ssum2=ssum2+z(j)
2150                 continue
                if (ivert eq 1) goto 2500
```

```
                if (lx lt ssuml) goto 2400
                xxg=ssuml*csafx
                if (lx eq ssuml) goto 2220
                call PLNSNK
                goto 2250
2220            continue
                ivert=1
                isym=3
2250            continue
                call WPLNPT(ifil)
                goto 3000
2400            continue
                xxg=lx*csafx
                ivert=1
                isym=3
                call WPLNPT(ifil)
2500            continue
                if (ssum2 ge lh) goto 3000
                if (ic eq ncx) goto 2520
                call PLNSNK
                goto 2600
2520            continue
                if (ib eq 2) goto 2530
                isym=0
                goto 2600
2530            continue
                if (ileg ne 2) goto 2540
                isym=4
                goto 2600
2540            continue
                isym=2
2600            continue
                call WPLNPT(ifil)

3000            continue
                xsum0=xsum1
                ssum0=ssum1
4000            continue


4100        continue
            zz=zout
            zmin=dmin1(zmin,zz)
```

```
            zmax=dmax1(zmax,zz)
            ncomp=ncomp+1
            npoint(ncomp)=npt
            zk= -zk
5000        continue

      zdoll=zero
      if (ileg eq 1) goto 6000
      xk= xxg*cospx
      zk= -zk+xxg*sinpx
      cospx=dcos(phih)
      sinpx=dsin(phih)
      zdoll=zk-xk*sinpx/cospx
      if (ifil eq 0) goto 5100
      igc=igc+1
      write(lu1,6) npoint(igc)
5100  continue
      npt=0
      xxg=zero
      isym=0
      call WPLNPT(ifil)
      if (l eq zero) goto 5200
      inafx=dcos(phih-phif)*inaf
      csafx=one/SECNT(inafx)
      xxg=l*csafx
      call WPLNPT(ifil)
5200  continue
      xsuml=x4
      call PLNCAT(1,ifil)
      ncomp=ncomp+1
      npoint(ncomp)=npt

6000  continue
      xx=xout
      zz=zout
      xmax=xx
      zmin=dmin1(zmin,zz)
      zmax=dmax1(zmax,zz)

      isym=7
      if (ifil eq 0) goto 6130
      igc=igc+1
      write(lu1,6) npoint(igc)
```

```
      xout=0 0
      zout=zdot1
      write(lu1,8) xout,zout,isym
      xout=xx
      zout=zz
      write(lu1,8) xout,zout,isym
 6130 continue
      ncomp=ncomp+1
      npoint(ncomp)=2
      call close(lu1)

 9000 continue
      return

    1 format(50a1)
    2 format(5a2)
    3 format(i2,' ',i2,' ',i2)
    4 format(i1)
    5 format(i2)
    6 format(i5)
    7 format(f8 2,1x,f8 2)
    8 format(f8 2,1x,f8 2,i2)
    9 format(19a1)
      end
*
```

```
e: sys final/12for/plnsnk for$$
      subroutine PLNSNK
*****************************************************************************
      implicit integer*2 (*)
      implicit double precision (a-z)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1
      real xout,zout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint(5)
      common /VELVPT/cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1,xout,zout,
     &  isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      clmp=z(is+2)
      if (clmp ge zero) goto 10
      isym=4
      goto 100
   10 continue
      if (clmp gt zero) goto 20
      isym=1
      goto 100
   20 continue
      isym=2
  100 continue
      return
      end
*
```

```
e! sys final/t2for/wplnpt for##
      subroutine WPLNPT(ifil)
*****************************************************************************
      implicit integer*2 (*)

      integer*2 ifil

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &    pref1,dum1,ext1,ext2

      double precision cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1
      real xout,zout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint(5)
      common /VELVPT/cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1,xout,zout,
     &  isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      xout=xk+xxg*cospx
      zout=zk+xxg*sinpx
      if (ifil eq 0) goto 100
      write(lu1,8) xout,zout,isym
  100 continue
      npt=npt+1
      return

    8 format(f8.2,1x,f8.2,i2)
      end
*
```

```
e! sys final/i2for/plncat for##
      subroutine PLNCAT(iris,ifil)
*****************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 iris,ifil

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1
      real xout,zout
      integer*2 isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint(5)
      common /VELVPT/cospx,sinpx,xk,zk,xxg,lx,zmin,zmax,
     &  ssum0,ssum1,xsum0,xsum1,xout,zout,
     &  isym,npt,ib,ncx,ioff,ic,ix,is,ncomp,npoint
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      isym=0
      xxg=xsum1
      if (iris ne 1) goto 1550
      isym=3
      goto 1580
 1550 continue
      if (ic eq ncx) goto 1560
      call PLNSNK
      goto 1580
 1560 continue
      if (ileg ne 1) goto 1570
      isym=3
      goto 1580
 1570 continue
      if (ib ne 2) goto 1580
```

```
      if (ileg ne 2) goto 1575
      isym=1
      goto 1580
 1575 continue
      isym=2
 1580 continue
      call WPIXPT(ifil)

      return
      end
```

```
ef sys final/f2for/moor04 for##
      program MOOR04
*************************************************************************
      implicit integer*2 (#)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncpl
      integer*1 pref1(21),dum1,ext1(4),ext2(4),ext3(4),ext4(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncpl,
     &  pref1,dum1,ext1,ext2,ext3,ext4

      integer*2 gbuff(24),lugraf,luprfl,ludbug
      common /GCB/ gbuff,lugraf,luprfl,ludbug

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VCLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      integer*2 npoint
      real hmin,hmax,hsym
      common /VHXCRV/ hmin,hmax,hsym,npoint

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision tnaf,phif
      common /VOFLR/ tnaf,phif

      double precision delyk,twod,halfd,dsq
      common /VANCH/ delyk,twod,halfd,dsq

      integer*2 ilib,ikey,iov
      integer*1 ans(1)

      integer*1 yes
      data yes/'Y'/
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
      call bfac1(0,'M40LY ')
      ilib=1
```

```
      ikey=1

  100 continue
      call ovlink('QUERY ',,ilib,ikey,iov,0,1,0,1)
      ilib=0
      call ovlink('PRSLV ')
      call ovlink('HXCALC ')
      write(screen,*) 'Do you want to compute another curve?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 100
      stop
      end
*
```

```
e! sys final/f2for/hxcalc for$$
      subroutine HXCALC
*************************************************************************
      implicit integer*2 (i)

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz

      integer*2 iscopa,iscopb,iiana,iianb,ii,is
      double precision epsy,gamma,se
      common /VCMPD/ epsy,gamma,se,iscopa,iscopb,iiana,iianb,ii,is

      integer*2 iiold
      double precision ss0,dten0,ss1,dten1,ss2,dten2,slp0,sa0,smin(2)
      common /VEQUAL/ ss0,dten0,ss1,dten1,ss2,dten2,slp0,sa0,smin,
     &  iiold
      equivalence (smin(1),samin),(smin(2),sbmin)

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      double precision snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
      common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh,dsnph

      double precision hinafh,hw4,w4h,s4w4h,c3h
      common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

      double precision epsxz,xziru(2),xzbas(2),hbas(2),scrat1(10)
      common /VCSSXZ/epsxz,xziru,xzbas,hbas,scrat1
      double precision xiru,ziru,xbas,zbas,hbasx,hbasz
      equivalence (xziru(1),xiru),(xziru(2),ziru),
     &  (xzbas(1),xbas),(xzbas(2),zbas),
     &  (hbas(1),hbasx),(hbas(2),hbasz)

      integer*2 iiant
      double precision a,b,snphi,tnafa,tnafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy
```

```
      common /VCSSHP/ a,b,snphi,inafa,inafb,
     &  seca7,seca8,ut,st,ykt,zkt,eex,eez,eey,ybuoy,itant

      integer*2 ivs
      double precision v0,v1,v2,f0,f1,f2,f,eps
      common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
      double precision varray(3),farray(3)
      equivalence (v0,varray),(f0,farray)

      integer*2 ilh0,ilh1,ilh2,il
      double precision lh0,lh1,lh2,ce
      common /VSCOIL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
      integer*2 ilh(3)
      double precision lh(3)
      equivalence (ilh,ilh0),(lh,lh0)

      double precision xred
      integer*2 isidf,nerra,nerrb
      common /VSTAB/ xred,isidf,nerra,nerrb
*******************************************************************************
* BEGIN EXECUTABLE CODE
*******************************************************************************
      if (ileg ne 1) goto 100
      call ovlink('HXCLC1 ')
      goto 200
  100 continue
      call ovlink('CPREP0 ')
      if (ileg eq 3) call CPREP1
      call CPREP2
      call ovlink('HXCLC2 ')
  200 continue
      return
      end
*
```

```
e! sys final/t2for/hxclc1 for##
      subroutine HXCLC1
*****************************************************************************
      implicit integer*2 (*)
      implicit double precision (a-z)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &  pref1,dum1,ext1,ext2

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),sla),(za(4),wla),(za(5),cla),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za 24),la),(za(25),phia)
      double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),slb),(zb(4),wlb),(zb(5),clb),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
```

```
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtol),(z(65),xtol),(z(66),ztol),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)

      integer*2 npoint
      real hmin,hmax,hsym
      common /VHXCRV/ hmin,hmax,hsym,npoint

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      integer*2 ncomp,npt,k,isg,iws
      real xmax,xmin,xcoord,delh,dhmax,hcoord
      equivalence (ncomp,npt,k),(dhmax,hcoord)
*****************************************************************************
* BEGIN EXECUTABLE CODE
*****************************************************************************
      eps=ya*1.0d-10
      delh=(hmax-hmin)/(npoint-1)

      ha=hmax*1.0d3
      call SECIV(nca,za,sinb,cosb,tanb,secb,eps,0)
      xmax=xa
      write(lu1,7) xmax,hmax

      if (hmin.eq.0.0) goto 120
      ha=hmin*1.0d3
      call SECIV(nca,za,sinb,cosb,tanb,secb,eps,0)
      xmin=xa
      goto 190
  120 continue
```

```
      call SUMSC(nca,za,sa,ca)
      xmin=(sa-ya)*cosb/(one-sinb)
  190 continue
      write(lul,7) xmin,hmin

      ncomp=1
      write(lul,5) ncomp

      dhmax=delh*1.0e-2
      if (hsym ge hmin  and  hsym le hmax) goto 220
      isg= -1
      goto 255
  220 continue
      isg=0
      do 250 k=1,npoint
           if (abs(hsym-hmin-(k-1)*delh) ge dhmax) goto 250
           isg=k
           goto 255
  250      continue
  255 continue

      npt=npoint
      if (isg eq 0) npt=npoint+1
      write(lul,6) npt

      iws=0
      do 1000 k=1,npoint
           hcoord=hmin+(k-1)*delh
           if (iws ne 0  or  isg ne 0  or  hsym ge hcoord) goto 500
           ha=hsym*1.0d3
           call SECIV(nca,za,sinb,cosb,tanb,secb,eps,0)
           xcoord=xa
           isym=5
           write(lul,8) xcoord,hsym,isym
           iws=1
  500      continue
           isym=0
           if (isg eq k) isym=5
           if (k ne 1) goto 550
           xcoord=xmin
           goto 900
  550      continue
           if (k ne npoint) goto 600
```

```
            xcoord=xmax
            goto 900
  600       continue
            ha=hcoord*1 0d3
            call SECIV(nca,za,sinb,cosb,tanb,secb,eps,0)
            xcoord=xa
  900       continue
            write(lu1,8) xcoord,hcoord,isym
            if (k ne 100*(k/100)) goto 1000
            write(screen,10) k
 1000       continue

      call close(lu1)
      return

    5 format(i2)
    6 format(i5)
    7 format(f8 2,1x,f8 2)
    8 format(f8 2,1x,f8 2,i2)
   10 format(1x,'JUST COMPLETED POINT',i4)
      end
*
```

```
e! sys final/!2for/hxclc2 for##
      subroutine HXCLC2
*************************************************************************
      implicit integer*2 (#)
      implicit double precision (a-z)

      integer*2 screen,keybd,lu1,lu2,niv99,siz99,ncp1
      integer*1 pref1(21),dum1,ext1(4),ext2(4)
      common /LUNITS/ screen,keybd,lu1,lu2,niv99,siz99,ncp1,
     &  pref1,dum1,ext1,ext2

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
      double precision za(25),zb(25)
      equivalence (z(1),za(1)),(z(26),zb(1))
      double precision ha,ala,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
     &  xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
     &  tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
      equivalence (za(1),ha),(za(2),ala,va),
     &  (za(3),s1a),(za(4),w1a),(za(5),c1a),
     &  (za(6),s2a),(za(7),w2a),(za(8),c2a),
     &  (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
     &  (za(13),x1a),(za(14),x2a),(za(15),x3a),
     &  (za(16),y1a),(za(17),y2a),(za(18),y3a),
     &  (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
     &  (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
      double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
     &  xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
     &  tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
      equivalence (zb(1),hb),(zb(2),alb,vb),
     &  (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
     &  (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
     &  (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
     &  (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
     &  (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
     &  (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
     &  (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
      double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
     &  h,phih,rtot,xtot,ztot,do
      equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
     &  (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
```

```
     &  (z(59),tana7),(z(60),tana8),(z(61),l),
     &  (z(62),h),(z(63),phih),
     &  (z(64),rtol),(z(65),xtol),(z(66),ztol),(z(67),do)
      double precision b,sinb,cosb,tanb,secb
      equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
     &  (z(29),secb)

      integer*2 npoint
      real hmin,hmax,hsym
      common /VHXCRV/ hmin,hmax,hsym,npoint

      double precision pi,halfpi,degrad,raddeg,zero,one,half
      integer*2 izero,ione,itwo
      common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
     &  izero,ione,itwo

      double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
     &  eex0,eez0,eey0,a0,b0,phia0,phib0
      integer*2 icase
      common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
     &  eex0,eez0,eey0,a0,b0,phia0,phib0,
     &  icase

      integer*2 ncomp,npt,k,isg,iws,istart
      real xmax,xmin,xcoord,delh,dhmax,hcoord
      equivalence (ncomp,npt,k),(dhmax,hcoord)
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
      delh=(hmax-hmin)/(npoint-1)

      h=hmax*1.0d3
      call CSXHP(0)
      xmax=rtol
      write(lu1,7) xmax,hmax

      if (hmin eq 0.0) goto 120
      h=hmin*1.0d3
      call CSXHP(0)
      xmin=rtol
      goto 190
  120 continue
      h=delh*0.5e3
```

```
      h=dmin1(h,(ca+cb+c3+s4*w4)*1.0d-4)
      call CSXHP(0)
      xmin=rtot
      h=half*h
      call CSXHP(1)
      xmin=rtot+rtot-xmin
  190 continue
      write(lu1,7) xmin,hmin

      ncomp=1
      write(lu1,5) ncomp

      dhmax=delh*1.0e-2
      if (hsym ge hmin  and  hsym le hmax) goto 220
      isg= -1
      goto 255
  220 continue
      isg=0
      do 250 k=1,npoint
           if (abs(hsym-hmin-(k-1)*delh) ge dhmax) goto 250
           isg=k
           goto 255
  250      continue
  255 continue

      npt=npoint
      if (isg eq 0) npt=npoint+1
      write(lu1,6) npt

      iws=0
      do 1000 k=1,npoint
           hcoord=hmin+(k-1)*delh
           if (iws ne 0  or  isg ne 0  or  hsym ge hcoord) goto 500
           h=hsym*1.0d3
           call CSXHP(1)
           xcoord=rtot
           isym=5
           write(lu1,8) xcoord,hsym,isym
           iws=1
  500      continue
           isym=0
           if (isg eq k) isym=5
           if (k ne 1) goto 550
```

```
            xcoord=xmin
            goto 900
  550       continue
            if (k ne npoint) goto 600
            xcoord=xmax
            goto 900
  600       continue
            h=hcoord*1 0d3
            istart=0
            if (k ge 3) istart=1
            call CSXHP(istart)
            xcoord=riot
  900       continue
            write(lul,8) xcoord,hcoord,isym
            if (ileg eq 3  and  k ne 10*(k/10)) goto 1000
            write(screen,10) k
 1000       continue

      call close(lul)
      return

    5 format(i2)
    6 format(i5)
    7 format(f8 2,1x,f8 2)
    8 format(f8 2,1x,f8 2,i2)
   10 format(1x,'JUST COMPLETED POINT',i4)
      end
*
```

```
et sys final/12for/csxhp for##
      subroutine CSXHP(istart)
*******************************************************************************
      implicit integer*2 (#)

      integer*2 istart

      integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
      double precision z(67),cz,cx,d,ta,tb
      common /VGLOB/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
     &  isol,ibrnch,uz
*******************************************************************************
* BEGIN EXECUTABLE CODE
*******************************************************************************
      call CPREP3
      if (ileg eq 2) goto 100
      call CSSHP
      goto 500
  100 continue
      call CSEHP(istart)
  500 continue
      return
      end
*
```

```
e! sys final/!2for/moor05 for##
      program MOOR05
*************************************************************************
      implicit integer*2 (i-n,#)
      integer*1 ifile(32),ans(1),ititle(50),idate(10),itime(8)
      integer*2 igname(11),ldc(11),lvw(11),pvw(11),
     &  ivert(18),kips(18),yfeet(18),zfeet(18)
      dimension x(250),y(250),isym(250),lfile(16)
      equivalence (ifile(1),lfile(1))
      integer*2 ifirst,ingo
      integer*1 pref1(21),ext2(4),ext3(4),ext4(4),blank,slash,yes
      data ext2/' LOC'/,ext3/' ELV'/,ext4/' PLN'/,blank/' '/,slash/'/'/
      data yes/'Y'/,keybd/9/,iscren/10/,lu/8/
      data ldc/'LOAD DEFLECTION CURVE '/,lvw/'ELEVATION VIEW          '/,
     &  pvw/'PLAN VIEW                '/
      data kips/'Vertical Axis  h in kips divided by '/
      data yfeet/'Vertical Axis  y in feet divided by '/
      data zfeet/'Vertical Axis  z in feet divided by '/
*************************************************************************
* BEGIN EXECUTABLE CODE
*************************************************************************
* Assign display screen to lu 10
*************************************************************************
      call assign('dc ',10)
      call glu(iscren)
*************************************************************************
* Read user input file
*************************************************************************
      call chrsiz(3)
      call erase
      write(iscren,*) 'Enter library name '
      read(keybd,*) pref1
      do 5 i=1,21
           j=22-i
           if (pref1(j) eq blank) goto 5
           ncp1=j+1
           pref1(ncp1)=slash
           goto 6
    5      continue
      ncp1=0
    6 continue
      do 8 i=1,32
           ifile(i)=blank
```

```
 8      continue
   if (ncpl eq 0) goto 10
   do 9 i=1,ncpl
        ifile(i)=prefl(i)
 9      continue

   ifirst=1
10 continue
   write(iscren,*) 'Enter file name '
   j=31-ncpl
   read(keybd,*) ifile(ncpl+1) j
   write(iscren,*) 'Enter graph type '
   write(iscren,*) '    1 - load displacement curve'
   write(iscren,*) '    2 - elevation view'
   write(iscren,*) '    3 - plan view'
   read(keybd,*) igraph
   if (igraph ne 1) goto 11
   call ADDEXT(ifile,31,ext2)
   goto 13
11 continue
   if (igraph ne 2) goto 12
   call ADDEXT(ifile,31,ext3)
   goto 13
12 continue
   call ADDEXT(ifile,31,ext4)
13 continue
   call file(1file,lu,2,istat)
   if (istat eq 0) go to 15
   write(iscren,14)(ifile(i),i=1,30),lu,istat
14 format(1x,30a1,i3,i3)
   go to 400

15 continue
   read(lu,18) igraph
18 format(i1)
   read(lu,19) ititle
19 format(50a1)
   read(lu,33) idate
33 format(10a1)
   read(lu,34) itime
34 format(8a1)
   read(lu,*) xxmax,yymax
   read(lu,*) xxmin,yymin
```

```
      read(lu,*) ncomp
*****************************************************************************
* Initialize titles depending on type of graph
*****************************************************************************
      goto (20,23,26), igraph
   20 continue
      do 21 i=1,11
            igname(i)=idc(i)
   21       continue
      do 22 i=1,18
            ivert(i)=kips(i)
   22       continue
      goto 29
   23 continue
      do 24 i=1,11
            igname(i)=ivw(i)
   24       continue
      do 25 i=1,18
            ivert(i)=yfeet(i)
   25       continue
      goto 29
   26 continue
      do 27 i=1,11
            igname(i)=pvw(i)
   27       continue
      do 28 i=1,18
            ivert(i)=zfeet(i)
   28       continue
   29 continue
*****************************************************************************
* Display first 7 records for verification
*****************************************************************************
      ingo=0
   30 continue
      call chrsiz(4)
      call hibrn8(10)
      write(iscrn,31)(igname(i),i=1,11)
   31 format(///5x,11a2)
      call chrsiz(3)
      call hibrn8(10)
      write(iscrn,*)'  output title   ',ititle
      write(iscrn,*)'  date    ',idate,'      time    ',itime
      write(iscrn,*)'  # of segments  ', ncomp
```

```
      write(iscren,*)'  xmin  ',xxmin,'      xmax  ',xxmax
      write(iscren,*)'  ymin  ',yymin,'      ymax  ',yymax
*******************************************************************************
* Allow user to alter x,y min/max
*******************************************************************************
      if (ifirst eq 1  or  ingo eq 1) goto 35
      write(iscren,*)
      write(iscren,*) 'Do you want to use your previous selection of gra
     &ph options?'
      read(keybd,*) ans
      if (ans(1) eq yes) goto 45
   35 continue
      write(iscren,*)
      write(iscren,*)'enter desired xmin '
      read(keybd,*) xmin
      write(iscren,*)'enter desired xmax '
      read(keybd,*) xmax
      write(iscren,*)'enter desired ymin '
      read(keybd,*) ymin
      write(iscren,*)'enter desired ymax '
      read(keybd,*) ymax
*******************************************************************************
* User enters step, scaling & ticks for x,y
*******************************************************************************
      write(iscren,*)'enter step size for x axis '
      read(keybd,*) xstep
      write(iscren,*)'enter scaling factor for x axis '
      read(keybd,*) ixscal
      write(iscren,*)'enter step size for y axis '
      read(keybd,*) ystep
      write(iscren,*)'enter scaling factor for y axis '
      read(keybd,*) iyscal
      write(iscren,*)'enter number of minor tick intervals per step for
     &x axis '
      read(keybd,*) ixtik
      write(iscren,*)'enter number of minor tick intervals per step for
     &y axis '
      read(keybd,*) iytik
      write(iscren,*)'do you want a grid? (y or n)'
      read(keybd,*) ans
      igrid=3
      if (ans(1) ne yes) igrid=0
      write(iscren,*)
```

```
      write(iscrn,*) 'Do you want to modify the graph options you have
     &just selected?'
      read(keybd,*) ans
      if (ans(1) ne yes) goto 40
      call erase
      ingo=1
      goto 30
   40 continue
*******************************************************************************
* Apply scaling factors
*******************************************************************************
      xmin=xmin/ixscal
      xmax=xmax/ixscal
      xstep=xstep/ixscal
      ymin=ymin/iyscal
      ymax=ymax/iyscal
      ystep=ystep/iyscal
*******************************************************************************
* Expand plot window boundaries to coincide with major ticks marks
*******************************************************************************
      xmin=xmin/xstep
      xym=int(xmin)
      if (xmin lt 0.0  and  xmin ne xym) xym=xym-1.0
      xmin=xym*xstep
      xmax=xmax/xstep
      xym=int(xmax)
      if (xmax gt 0.0  and  xmax ne xym) xym=xym+1.0
      xmax=xym*xstep
      ymin=ymin/ystep
      xym=int(ymin)
      if (ymin lt 0.0  and  ymin ne xym) xym=xym-1.0
      ymin=xym*ystep
      ymax=ymax/ystep
      xym=int(ymax)
      if (ymax gt 0.0  and  ymax ne xym) xym=xym+1.0
      ymax=xym*ystep
*******************************************************************************
* Erase screen and write titles on screen
*******************************************************************************
   45 continue
      ifirst=0
      call erase
      write(iscrn,1)(idate(i),i=1,10),(igname(i),i=1,11),
```

```
     &  (ttime(i),i=1,8)
      call chrsiz(4)
      call hibrn8(10)
      write(iscren,2)(ititle(i),i=1,50)
      call chrsiz(3)
      call hibrn8(10)
      write(iscren,3)(iver1(i),i=1,18),iyscal,ixscal
    1 format(1x,'Date   ',10a1,25x,11a2,25x,'Time   ',8a1)
    2 format(/25x,50a1/)
    3 format(1x,18a2,i4,26x,
     &  'Horizontal Axis  x in feet divided by ',i4)
*************************************************************************
* Draw axis,labels, tick marks & grid by plotting a dummy data point
*************************************************************************
      call init
      call page(-30500,32000,-24000,23000)
      call line(1)
      call grid(igrid)
      call xticks(ixtik)
      call yticks(iytik)
      call xylin(xmin,xstep,xmax,ymin,ystep,ymax)
      call xlab(1,4,-1,3)
      call ylab(1,4,-1,3)
      call plot(xmin,1,ymin,1,1)
*************************************************************************
* Iteration to plot all segments including ocean floor
* Npoint is the # of data points within one segment
*************************************************************************
      do 250 i=1,ncomp
           read(lu,*) npoint
           n=0
   50      continue
           isave=0
           if (npoint le 250) go to 60
           isave=npoint-249
           npoint=250
   60      continue
*************************************************************************
* Read data points and symbol value, normalize x,y with scaling factor
*************************************************************************
           npt=npoint-n
  100      continue
           do 150 j=1,npt
```

```
                k=j+n
                read(lu,*) x(k),y(k),isym(k)
                x(k)=x(k)/ixscal
                y(k)=y(k)/iyscal
  150           continue
*****************************************************************************
* Plot data points, symbol 7 represents ocean floor
*****************************************************************************
            if (isym(1) eq 7) call line(4)
            call symbol(0)
            call plot2(x(1),1,y(1),1,npoint)
            if (isym(1) eq 7) go to 250
*****************************************************************************
* Place symbols on line just drawn,
* save y data point for ocean surface if buoy (sym = 5)
*****************************************************************************
            do 200 j=1,npoint
                isymj=isym(j)
                if (isymj eq 0) goto 200
                call symbol(isymj)
                call plot2(x(j),1,y(j),1,1)
                if (isymj eq 3 and igraph eq 2) ysave=y(j)
  200           continue
*****************************************************************************
* If more than 250 data points, save last x,y,symbol
* Reset npoint to remaining # of points
*****************************************************************************
            if (isave eq 0) go to 250
            npoint=isave
            n=1
            x(1)=x(250)
            y(1)=y(250)
            isym(1)=0
            go to 50
  250       continue
*****************************************************************************
* If elevation view, draw ocean surface
*****************************************************************************
      if (igraph ne 2) go to 300
      call line(3)
      x(1)=xmin
      y(1)=ysave
      x(2)=xmax
```

```
      y(2)=ysave
      call plot2(x(1),1,y(1),1,2)
  300 continue
*****************************************************************************
* Display and frame graph
*****************************************************************************
      call frame
*****************************************************************************
* Replot same file?
*****************************************************************************
      call readfk(keys)
*     if (keys eq 0) go to 400
      call erase
      write(iscren,*)'do you wish to plot this file again? (y or n)'
      read(keybd,*) ans
      if (ans(1) ne yes) go to 350
      rewind lu
      go to 15
*****************************************************************************
* Plot a different file?
*****************************************************************************
  350 continue
      write(iscren,*)'do you wish to plot another file? (y or n)'
      read(keybd,*) ans
      if (ans(1) ne yes) go to 400
      call close(lu)
      go to 10

  400 continue
      call close(lu)
      stop
      end
*
```

END

DATE
FILMED

5-83

DTIC